

Graph Analysis of Delay in MVC Codecs with Limited Processing Capacity

Pablo Carballeira, Julián Cabrera, Fernando Jaureguizar and Narciso García

Abstract-- We present an extension of the graph theory analysis of encoding/decoding delay in multiview video coding schemes for hardware devices with limited processing capacity. With this, we provide a unified methodology for the analysis of multiview delay in device models with unlimited and limited processing capacity.

I. INTRODUCTION

Video technologies have targeted for years the development of systems that provide immersive viewing experiences. 3D Video (3DV) and Free Viewpoint Video (FVV) systems expand the user's experience beyond 2D video [1], providing a 3D depth impression of the scene, and interactive viewpoint selection. To support 3DV and FVV, several coding schemes for multiview video and additional data, such as MVC [2], 3D-AVC [3] or extensions of HEVC [4], have been developed and standardized. All of these coding schemes include multiview video as the fundamental element of the data that is coded and transmitted. The core of the multiview coding technology has consistently been the extension of hybrid video coding schemes to the spatial dimension. Thus, complex prediction structures are constructed, with prediction relationships in the temporal and spatial dimensions.

In previous works [5] [6], we provided a framework for the analysis of delay in multiview video coding schemes. This framework provides models based on graph theory that assume unlimited (or sufficient) processing resources in the codec devices. To complete this framework, we have also developed other type of models that cover the analysis in device hardware architectures with limited processing capacity, with a level of generality that make them applicable for a good range of implementations: the Fixed multi-processor (MP) model [5], the Flexible MP model [5] and the Parallel MP model [6]. While the Parallel MP model fits into the graph theory formulation [6], the Fixed and Flexible MP models were developed by simulation of the encoding and decoding processes.

Here, we investigate a novel approach to analyze the delay, within the formulation of graph theory, for devices with limited processing capacity. This is done by an extension of the graph under analysis, in which the new edges represent the queuing dependencies imposed by the limitations of the processing resources. By this extension of the graph theory framework, we provide a systematic analysis of the delay in codec devices with limited processing capacity, for which we

are able to identify the delay added by each dependency (prediction and queuing), compute the delay of each frame, or identify delay bottlenecks. Thus, we are able to apply this knowledge in the design of multiview video applications with low-delay requirements, i.e. defining the prediction structure [5] or the processing resources [6], in scenarios that consider hardware device models with the processing limitations of consumer devices. In this paper, we show the methodology for the extension of the graphs to meet the delay performance of the Fixed MP model, leaving as future work the extension towards more general models such as the Flexible MP model.

II. EXTENSION OF THE GRAPH

Here, we describe the approach to extend the initial graph for the analysis of delay in codec devices with limited processing capacity. As explained in [5], the edges of the initial graph indicate prediction dependencies between frames. Thus, their cost represents the delay added by a parent frame (reference frame) to the encoding/decoding process of a child frame. Given only those prediction dependencies, it is still possible that two frames are ready to be encoded/decoded at the same instant. Thus, the new set of edges represents the delay added by one frame to another frame due to the occupation of processing resources that the first frame realizes during its processing time. In the paper, we first describe the cost of these new graph edges. Then, for the specific case of the Fixed MP model, we describe the algorithm for the inclusion of those new edges.

A. Cost of new edges

The cost value $\omega_{j,l}^{i,k}$ of the edge that links node x_j^i (j -th frame of view i) with node x_l^k (x_l^k occupies the processing resources that frame x_l^k precises) is:

$$\omega_{j,l}^{i,k} = (t_{\text{capt } j}^i + \Delta t_{\text{proc } j}^i) - t_{\text{capt } l}^k, \quad (1)$$

where $t_{\text{capt } j}^i$ and $t_{\text{capt } l}^k$ are the capture times of frames x_j^i and x_l^k respectively, and $\Delta t_{\text{proc } j}^i$ is the processing time of x_j^i .

B. Algorithm to include new edges: Fixed MP model

The graph is initialized with the edges and costs given by the prediction structure [5]. The graph includes the nodes corresponding to all the frames that are encoded/decoded (all GOPs), or the number of GOPs that are needed to compute a stable value of the encoding/decoding latency (see the examples in [5]). Then, iteratively, the processing chronogram

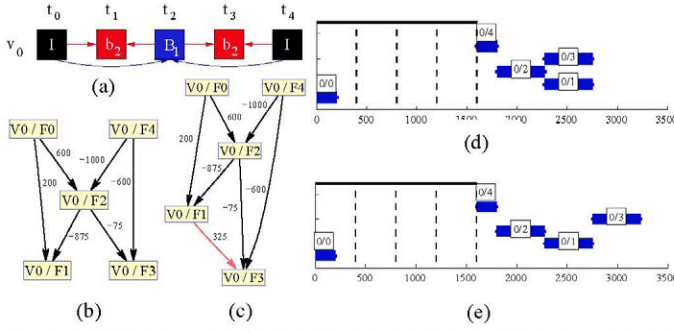


Fig. 1. (a) GOP 1 view 5 frames. (b) Initial graph. (c) Extended graph. (d) Initial encoding chronogram. (e) Final encoding chronogram. In the chronogram figures, the x-axis represents time and the processing time of each frame is labeled with v_i/t_j .

is computed and new edges are added for pairs of nodes that are concurrent in a given processor. In the following, we explain this process in detail:

1. Initialization of the DAG: it contains all frames of all GOPs, and the edges represent prediction dependencies.
2. The encoding/decoding chronogram is computed using the graph theory model (infinite number of processors).
3. In increasing order of $t_{start_j}^i$, do the operations per frame:
 - a. Find the set of frames $\{x_k^i\}$ (same view i), that are processed concurrently with frame x_j^i .
 - b. For each frame x_k^i : establish a new edge $x_j^i \rightarrow x_k^i$ with the cost described in (1).
 - c. Re-compute the chronogram using the graph with added edges and jump back to 3.a.
4. Finish when the three previous operations have been performed for all frames.

As an example, we provide the results obtained with this approach for a simple GOP of one view and five frames (Figure 1a). Figure 1b shows the initial graph obtained from that prediction structure. Figure 1d represents the encoding chronogram obtained for the graph in Figure 1b. Note that the processes of frames $V0/F1$ and $V0/F3$ are concurrent (this could not be possible in an encoder with one only processor). Figure 1c shows the extended graph with the new edge (red) that establishes a queuing dependency between frames $V0/F1$ and $V0/F3$. Figure 1e depicts the chronogram for the extended graph, in which there are no concurrent processes.

III. RESULTS

This method was applied for different prediction structures of the JMVM model and non-normative multiview prediction structures. The delay performance that was obtained is the same than that of the Fixed MP model [5], which is computed by analysis of encoding order, and thus, serves as ground truth results. As an example of the graph results obtained for the encoding process of multiple GOPs, Figure 2a shows the initial graph for three GOPs of the prediction structure in

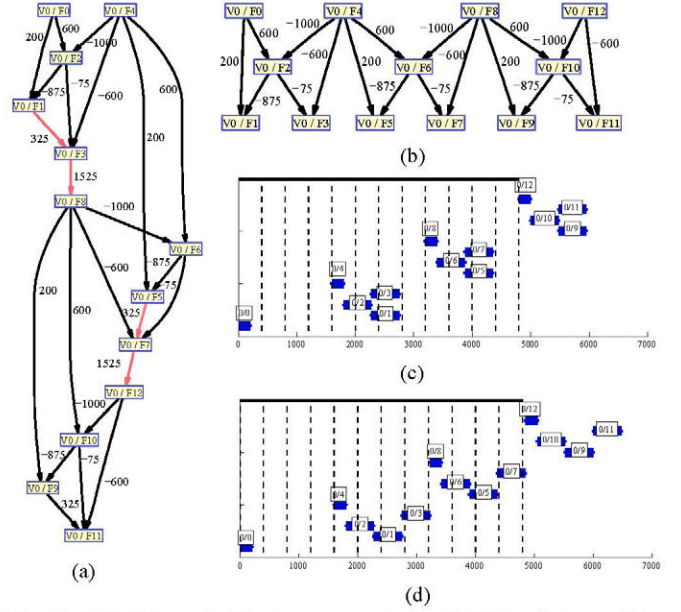


Fig. 2. (a) Initial graph. Graph correspondent to 3 GOPs of the prediction structure in Figure 1a (b) Extended graph. (c) Initial encoding chronogram. (d) Final encoding chronogram.

Figure 1a. Figure 2c shows the corresponding chronogram. Figure 2b shows the graph in Figure 2a extended with four new edges (red). Figure 2d shows the chronogram for the extended graph, in which there are no concurrent processes.

IV. CONCLUSIONS AND FUTURE WORK

We provided an extension of the graph theory framework for the analysis of the encoding/decoding delay to cover devices with limited processing capacity. Thus, we provide a way to apply the benefits of the delay analysis through graph theory to devices with limitations in processing resources such as consumer codec devices. Here, we have shown the extension of graphs with new edges that represent the queuing dependencies in the processors for the case of the Fixed MP model. As future work, we will provide algorithms for more general hardware architectures as the Flexible MP model.

V. REFERENCES

- [1] A. Smolic, K. Müller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video - technologies, applications and MPEG standards," in Proc. of ICME, Jul. 2006, pp. 2161–2164.
- [2] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y. Wang, "Joint Draft 8.0 on Multiview Video Coding," Doc. JVT-AB204, 28th JVT meeting, Hannover, Germany, Jul. 2008.
- [3] ISO/IEC JTC1/SC29/WG11, "3D-AVC Test Model 7," Output doc. N13753, 105th MPEG meeting, Vienna, Austria, Aug. 2013.
- [4] G.J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE TCSVT*, vol.22, no.12, pp.1649-1668, Dec. 2012.
- [5] P. Carballera, J. Cabrera, A. Ortega, F. Jaureguizar, and N. Garcia, "A framework for the analysis and optimization of encoding latency for multiview video," *IEEE JSTSP*, vol. 6, no. 5, pp. 583–596, Sep. 2012.
- [6] P. Carballera, J. Cabrera, F. Jaureguizar, and N. Garcia, "Systematic analysis of the decoding delay in multiview video," *Journal of Visual Comm. and Image Repr. on*, vol. 25, no. 4, pp. 689–697, May 2014.
- [7] P. Carballera, J. Cabrera, F. Jaureguizar, and N. Garcia, "Systematic analysis of the decoding delay on MVC decoders," in Proc. of ICCE, Jan. 2012, pp. 243–244.